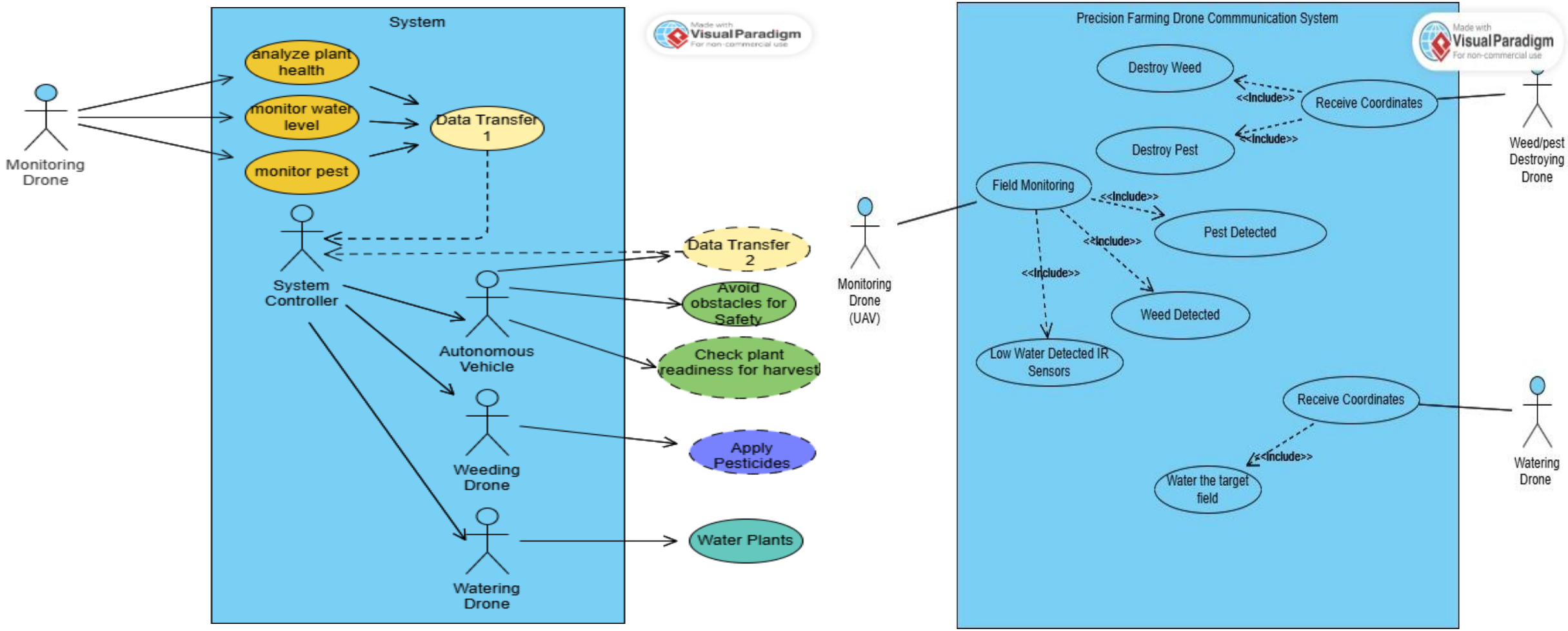
AUTONOMOUS SYSTEM B

GROUP-C

MUHAMMAD UMER BIN YAQOOB
AMMAR KHAN
SHAFIN

PRECISION FARMING



Drones

Watering

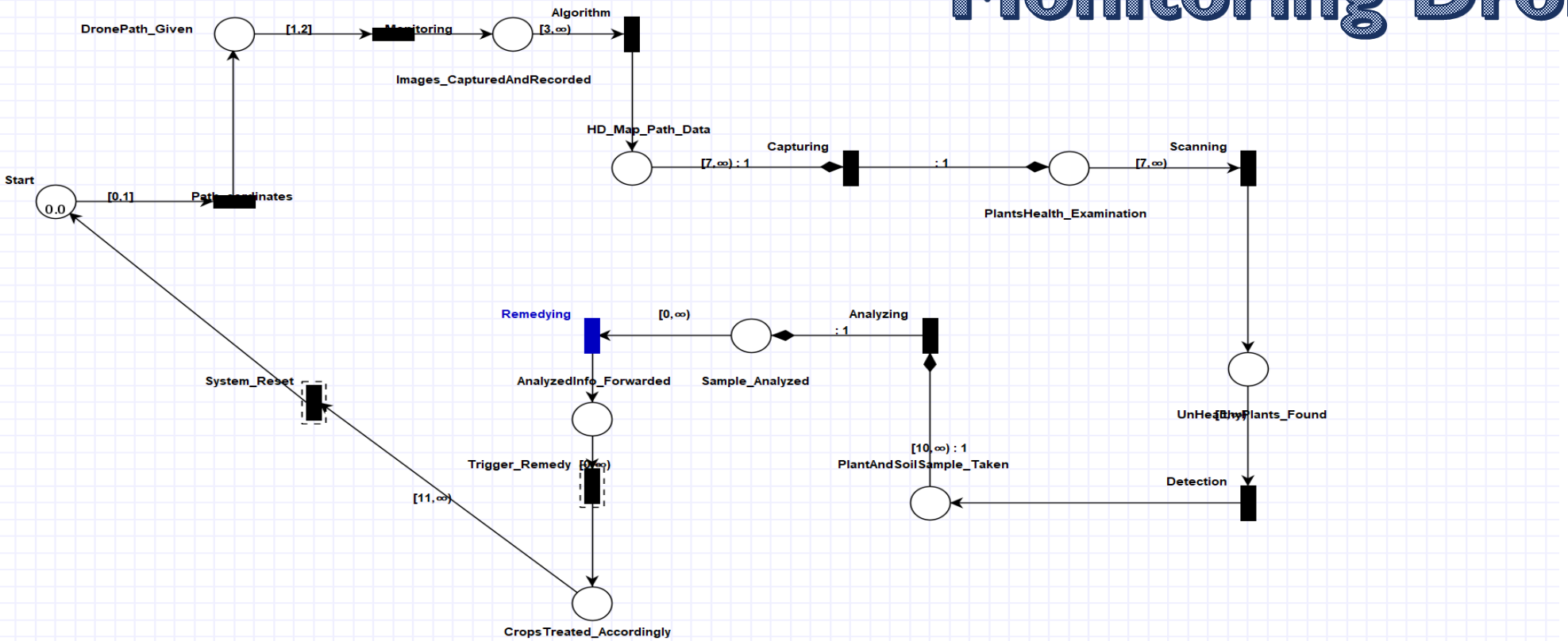
Monitoring

Weeding

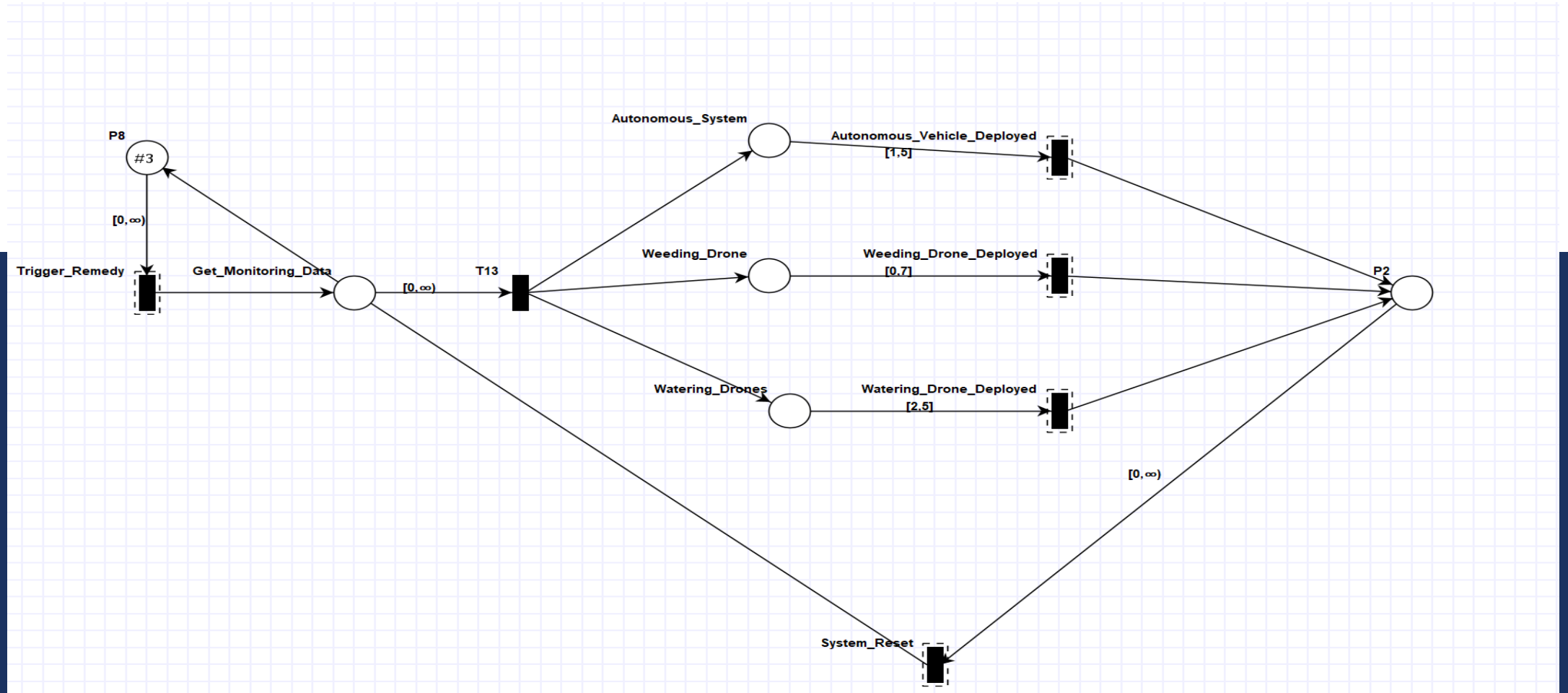
1-MONITORING DRONE
2-WEEDING DRONE
3-WATERING DRONE

TAPAAL MODELS

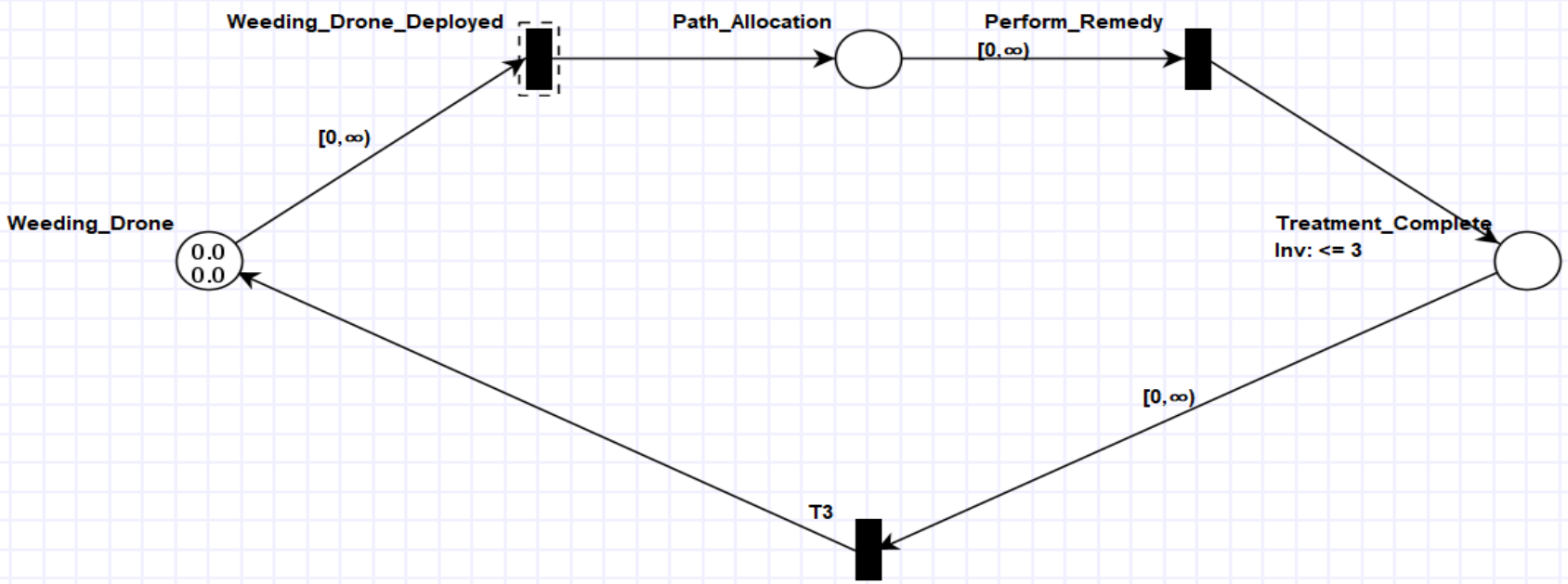
Monitoring Drone



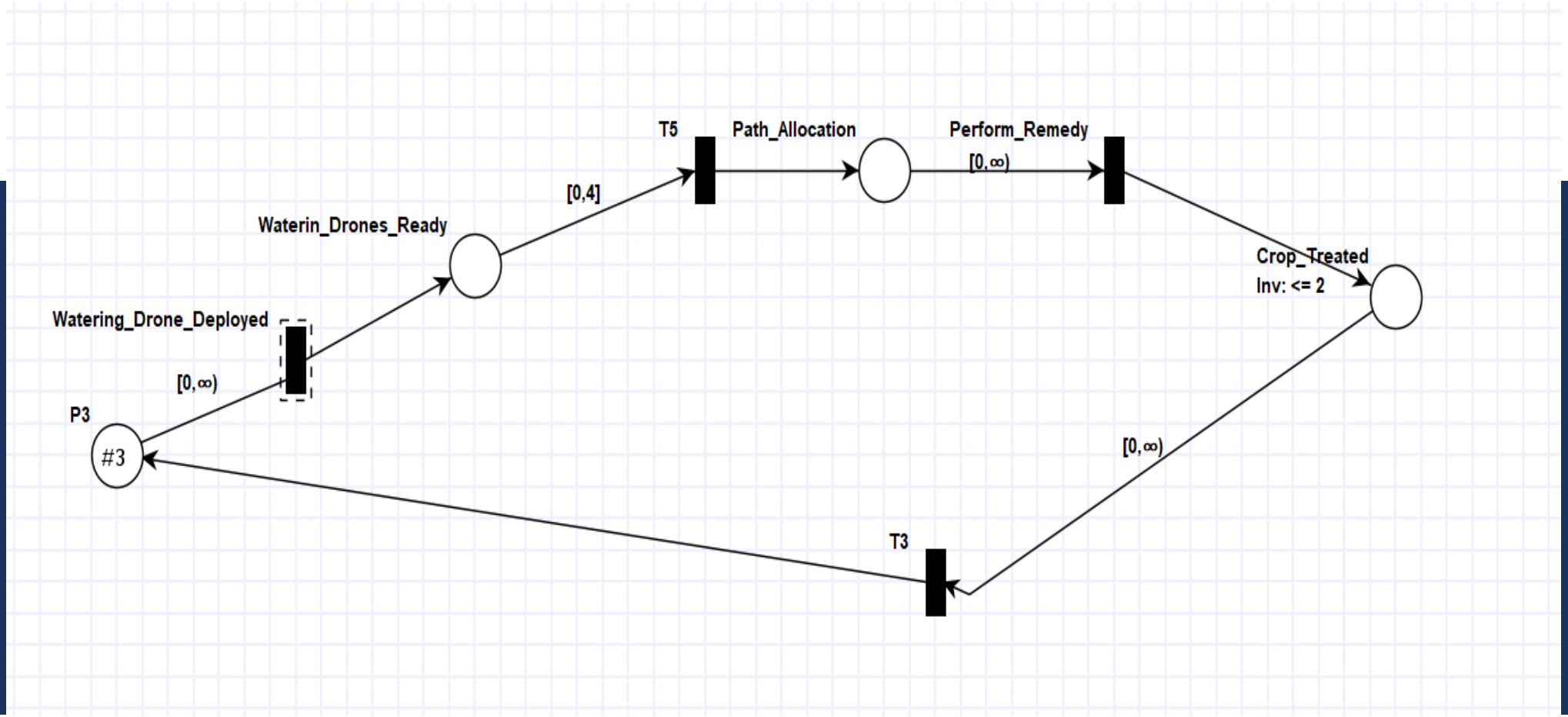
Controller


P8
#3
[0,∞)
Trigger_Remedy
Get_Monitoring_Data
[0,∞)
T13
Autonomous_System
Autonomous_Vehicle_Deployed
[1,5]
Weeding_Drone
Weeding_Drone_Deployed
[0,7]
Watering_Drones
Watering_Drone_Deployed
[2,5]
P2
[0,∞)
System_Reset

WEED REMOVING Drone



Watering Drone



RESNET 50

Input: Image resized to 224×224×3

Zero Padding: Preserves image dimensions

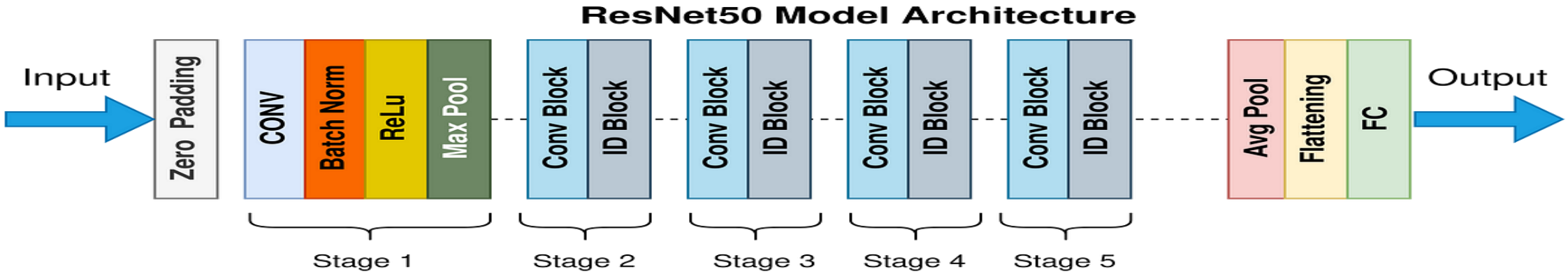
Stage 1: Convolution → BatchNorm → ReLU → MaxPool.

Stages 2–5: Convolutional + Identity Blocks with Skip

Connections (deepen feature learning).

Global Average Pooling: Converts feature maps to a vector.

Fully Connected Layer: Outputs class probabilities.



DATA LOADING

- Loads and splits dataset into training (80%) and validation (20%) using
- ImageDataLoaders.from_folder.

- Applies image resizing (224x224) and augmentations (flipping, rotation, zooming) for better model training.

- Visualizes a batch of images to inspect the preprocessing and augmentations.

```
# Create DataLoaders for the dataset
dls = ImageDataLoaders.from_folder(
    data_path,
    valid_pct=0.2,        # Use 20% of the data for validation
    seed=42,              # Seed for reproducibility
    item_tfms=Resize(224), # Resize images to 224x224 (ResNet requirement)
    batch_tfms=aug_transforms(do_flip=True, max_rotate=30, max_zoom=1.1) # Augmentations
)

# Show a batch of images
dls.show_batch(max_n=8, figsize=(8, 8))
```

MODEL TRAINING

- Creates a vision model using ResNet50 as the backbone with FastAI's vision_learner.
- Tracks accuracy during training as the evaluation metric.
- Finds the optimal learning rate with lr_find() to ensure faster and effective training.

```
# Create a learner with ResNet50
learn_plant = vision_learner(
    dls,
    resnet50,              # Use ResNet50 as the backbone
    metrics=accuracy       # Track accuracy during training
)

# Find the optimal learning rate
learn_plant.lr_find()
```

TRAINING AND VALIDATION

epoch	train_loss	valid_loss	accuracy	time
0	2.145392	0.849470	0.726781	07:47

epoch	train_loss	valid_loss	accuracy	time
0	1.071802	0.474022	0.840397	01:31
1	0.731097	0.363347	0.866546	01:30
2	0.527963	0.286733	0.891794	01:31
3	0.391679	0.240063	0.908927	01:29
4	0.347866	0.234843	0.908927	01:30

DEMO RUN

The model was run on the dataset of 900 plants and weeds images.

12 Different classes (e.g Fat-hen, Black Grass, Sugar Beet etc..)

RESULTS

```
280 Image335, (10, 9) to (11, 5), 5
281 Image348, (10, 22) to (11, 18), 5
282 Image350, (11, 18) to (11, 20), 2
283 Image352, (11, 20) to (11, 22), 2
284 Image353, (11, 22) to (11, 23), 1
285 Image354, (11, 23) to (11, 24), 1
286 Image355, (11, 24) to (11, 25), 1
287 Image356, (11, 25) to (11, 26), 1
288 Image357, (11, 26) to (11, 27), 1
289 Image358, (11, 27) to (11, 28), 1
290 Image359, (11, 28) to (11, 29), 1
291 Image360, (11, 5) to (12, 0), 6
```

RemovedWeedwithCoordinates.txt

```
280 Image335, 309 to 335, 5
281 Image348, 322 to 348, 5
282 Image350, 348 to 350, 2
283 Image352, 350 to 352, 2
284 Image353, 352 to 353, 1
285 Image354, 353 to 354, 1
286 Image355, 354 to 355, 1
287 Image356, 355 to 356, 1
288 Image357, 356 to 357, 1
289 Image358, 357 to 358, 1
290 Image359, 358 to 359, 1
291 Image360, 335 to 360, 6
```

removed.txt